

---

## 2. Bit Packing

**Program Name:** BitPacking.java

**Input File:** bitpacking.dat

A small specialty store in town wants you to write a program that can read in data from their specialized inventory database and print out the information in human readable format so they know what they have in stock and what they need to order. Because they are charged by the size of their database, they have condensed their data into single 32 bit integers whose bits contain 3 unique pieces of information: the item's type, the item's sub-type, and the amount of the item left in stock. The data is stored as follows:

- The top 9 bits are currently unused, and can be assumed to be 0.
- The next 5 bits contain the inventory item's type, will be a value from 1 to 26, and will correspond to the uppercase letters of the alphabet, A through Z, in order.
- The following 6 bits contain the inventory item's sub-type and will range in value from 1 to 61, where 1 to 26 are the uppercase letters A through Z, 27 through 52 are the lowercase letters a through z, and the values 53 through 61 are the numbers 1 through 9.
- The final bits represent the amount of the item they have in inventory.

### Input

The first line of input will contain a single integer  $n$  that indicates the number of packed bit values that will follow. Each of the following  $n$  lines will contain a single integer (in base 10) representing a packed bit value.

Take, for example the first input value 536839, converted to binary and cut up into 4 groups using the problem description:

```
0000000000 00010 000011 000100000111
      0         2       3       263
```

The first group of nine bits converts to decimal 0 and is unused. The second group of 5 bits is type 2, corresponding to B; the third group of 6 bits is subtype 3, corresponding to C; and the fourth and final group of bits is 263, the amount of inventory they have.

### Output

For each packed bit value you will print out a single line in the form

```
Type: X | Sub Type: Y | Inventory: Z
```

Where  $X$  is the upper case character denoting the type,  $Y$  is the upper case, lower case or numerical value of the sub-type, and  $Z$  is the amount they have in inventory, in base 10.

### Example Input File

```
2
```

```
536839
```

```
6672394
```

### Example Output to Screen

```
Type: B | Sub Type: C | Inventory: 263
```

```
Type: Y | Sub Type: c | Inventory: 10
```